```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$                                                                                       $
$                                                                                       $
$     *************************************************************************         $
$     *******************  TOUGHREACT  October 2004    ************************         $
$     *************************************************************************         $
$                                                                                       $
$     .......READ.ME..........READ.ME..........READ.ME..........READ.ME........         $
$     .........................................................................         $
$     .............................Update 10/04................................         $
$     .........................................................................         $
$                                                                                       $
$                                                                                       $
$        TOUGHREACT is a program for non-isothermal multiphase reactive fluid           $
$     flow and geochemical transport in porous media and fractured rocks. The code      $
$     is written in Fortran 77. It is developed by introducing reactive geochemical     $
$     transport into the framework of the existing multi-phase fluid and heat           $
$     flow code TOUGH2 V2 (Pruess et al., 1999).                                        $
$                                                                                       $
$                 Developed by                                                          $
$                             Tianfu Xu                                                 $
$                             Eric Sonnenthal                                           $
$                             Nicolas Spycher                                           $
$                             Karsten Pruess                                            $
$                                                                                       $
$                           At Lawrence Berkeley National Laboratory                    $
$                                                                                       $
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

This flyer contains brief instructions for installing and running TOUGHREACT

TOUGHREACT is distributed on CD. Because TOUGHREACT was derived from TOUGH2 V2,in addition to the current manual, users must have the manual of the TOUGH2 V2 (Pruess et al., 1999). Information on TOUGH2 V2 is also available on the TOUGH2 homepage on the web.

```
*************************************************************************
*               http://www-esd.lbl.gov/TOUGH2/                          *
*************************************************************************
```

===

1. THIS CD INCLUDES FIVE SUBDIRECTORIES:

1.1. Subdirectory: ‘documents’ contains the manual of the TOUGHREACT code.

1.2. Subdirectory: ‘source-files’ contains source files of the code:

```
treact.f       - This module contains the main program of TOUGHREACT, and
                 must be compiled with INCLUDE files ‘T2’, ‘chempar23.inc’,
                 common23.inc, and ‘perm23.inc’ with PARAMETER statements
                 for flexible dimensioning of all major arrays;

T2             - an INCLUDE file for PARAMETER statements of fluid and
                 heat flow;

chempar23.inc   - an INCLUDE file for PARAMETER statements of reactive
                 geochemical transport;

common23.inc   - an INCLUDE file for common blocks for reactive
                 geochemical transport;

perm23.inc     - an INCLUDE file for PARAMETER statements related to porosity
                 and permeability changes due to mineral dissolution and
                 precipitation;
```

```
t2cg22.f       - includes an executive routine LINEQ for the linear equation
                 solution, which may call a solver package t2solv.f. It also
                 includes routines for preconditioning of the Jacobian matrix;

t2solv.f       - a package of conjugate gradient solvers;

meshm.f        - module with internal mesh generation facilities;

t2f.f          - the core module of TOUGH2; it reads input data, initial-
                 izes arrays and parameters, sets up the Jacobian matrix
                 for Newton-Raphson iteration, and performs time stepping.
                 It also contains the water property routines (steam table
                 equations) and the relative permeability and capillary
                 pressure functions which are used in the equation-of-state
                 modules;

multi.f        - formulate Jacobian matrix and right-hand side residual
                 terms for multi-phase fluid and heat flow;

readsolu.f     - read and initialize data related to solute transport;

inichm.f       - read and initialize data related to geochemistry;

geochem.f      - subroutines for geochemical calculations and writing aqueous
                 concentrations and mineral abundances at specified
                 time steps;

newton.f       - assemble Jacobian matrix of Newton-Raphson iteration for
                 solving equations of geochemical system;

rctprop.f      - calculate (1) reactive surface areas for reactions of mineral
                 dissolution and precipitation, (2) porosity and permeability
                 changes due to mineral dissolution and precipitation,
                 and (3) Leverett scaling for capillary pressure modification
                 due to porosity and permeability changes;

ma28abc.f      - MA28 linear equation solver is not available for TOUGHREACT
                 unlike TOUGH2. TOUGHREACT uses only onjugate gradient (iterative)
                 solvers. To avoid unsatisfied externals, some dummy
                 subroutines are placed in file ma28abc.f.
```

Two versions of the TOUGHREACT source program are available on the distribution CD. One contains IMPLICIT DOUBLE PRECISION (A-H, O-Z) and IMPLICIT INTEGER*8 (I-N) statements that will automatically generate 64-bit arithmetic on 32 bit processors. Another contains the same code but without the INTEGER*8 statements, because most PC compilers do not recognize the *8 statements. The latter version may require special compliler options to generate 64-bit code at compile time. Therefore, two subdirectories: 'integerx8' and 'integer', are created coresponding to the two versions. Note that the original TOUGH2 V2.0 is intrinsically single-precision and requires special compliler options to generate 64-bit code at compile time.

Each version also includes three low-level subdirectories: (1) 'EOS-modules' (equation of state for fluid flow, see original TOUGH2 V2 manual; Pruess et al., 1999), (2) 'SECOND-subroutines' (CPU-time subroutine that is machine-dependent), and (3) 'makefiles' (used for compiling and linking, which is compiler-dependent).

'EOS-modules' contains:

```
       eos1.f   - equation of state for water, or two waters;
       eos2.f   - equation of state for water/CO2 mixtures;
       eos3.f   - equation of state for water/air;
       eos4.f   - equation of state for water/air (with vapor pressure lowering effects);
       eos9.f   - equation of state for saturated-unsaturated flow
                      (Richards' equation);
       eco2.f   - equation of state for water-brine-CO2 mixtures (for
                      CO2 deep saline aquifer disposal). This routine requires a
                      property data file CO2TAB is required, which can be found
                      in the current subdirectory.
```

'SECOND-subroutines' contains:

```
       second_pc.f   - timer subroutine for PC
       second_mac.f  - timer subroutine for Macintosh
       second_sun.f  - timer subroutine for SUN workstation
```

```
          second_dec.f  - timer subroutine for DEC ALPHA workstation
          second_ibm.f  - timer subroutine for IBM RISC System/6000 workstation,
                          running a UNIX-based operating system

   'makefile':
          makefile   for  DEC ALPHA Workstation, or
                          SUN SPARC Workstation, or
                          Intel Fortran Compiler on linux, or
                          Mac PPC - g77, or
                          IBM - f77
```

1.3. Subdirectory: 'sample-problems'

The directory contains a total of 8 subdirectories. Each subdirectory contains one sample problem. Simulations of sample problem presented here were run on Pentium 4 PC machines (1.7G). These EXE files were generated with COMPAQ Visual Fortran compiler version 6.6, and are also provided in the distribution CD. The descriptions of sample problems are given in Chapter 8 of the manual.

'P1_EOS9_kd-decay': Linear sorption and decay.

'P2_EOS9_Aquia-aquifer': Water quality in the Aquia aquifer.

'P3_EOS3_YM-calcite': Infiltration and calcite deposition at Yucca Mountain.

'P4_EOS4_Heat-test': Yucca Mountain heat test problem.

'P5_ECO2_1D-radial': CO2 disposal in deep aquifers.

'P6_EOS9_copper': Supergene copper enrichment.

'P7_EOS2_LVC': Caprock alteration in Long Valley Cadera (LVC).

'P8_EOS1_scaling': Reservoir mineral scaling by injection.

1.4. Subdirectory: 'EXE-files-PC'

This contains executable files with different fluid flow modules for running problems on PCs. The PC EXE files were generated with COMPAQ Visual Fortran compiler 6.6.

1.5. Subdirectory: 'utility-programs'

The 'utility-programs' directory provides utility programs for the chemical database. The descriptions of these programs are given in Appendix J of the TOUGHREACT manual. It includes four low-level subdirectories:

'convert-eq3/6' contains the source code and an example input for converting EQ3/6 database (data0) to TOUGHREACT database.

'switch-basis' contains the source code and an example input for switching basis (primary) species.

'regress-logK' contains the source code for regressing logK data.

'check-balance' contains the source code and an example input for checking mass and charge balances of reactions.

==========

2. CODE INSTALLATION

Step 1: Create a directory in user's computer

Step 2: Copy all files from the root of subdirectory 'source-files/integerx8'

Step 3: If necessary, adjust parameter statements (dimensioning variables)
in T2 (for fluid and heat flow dimension parameters) and
chempar23.inc file (for chemistry dimension parameters)
to be sufficient for the simulation problem.
For example, MNEL = 8000 (maximum number of grid blocks)
and MNCON = 20000 (maximum number of connections) are specified
in the INCLUDE file T2 with the distribution CD.

Step 4: Copy a desired EOS-module from the low-level subdirectory
'EOS-modules'. Note that only one of the EOS modules must
be linked at a time.

Step 5: Copy a timer subroutine from the low-level subdirectory
'SECOND-subroutines', depending on user's machine.

Step 6: Copy makefile from the low-level subdirectory 'makefile', and modify
the provided makefile depending on user's machine and compiler.
Makefile needs to change with corresponding EOS module
and timer subroutine. (For some PC Windows-based compilers,
see *Remarks below.)

Step 7: Type: 'make' (such as for SUN workstation)

*Remarks: for many PC Windows-based compilers, special compiler options may be
required to generate 64-bit code at compile time. For example, using COMPAQ
Visual Fortran compiler 6.6, the user needs to select proper window
boxs. The following steps must be followed:

(1) Developer Studio

Start DIGITAL/Compaq Visual Fortran's Developer Studio.

(2) Create new Project

Select <File>, <New...>
Select <Projects>, <Win32 Console Application>,
<Project name:> treact
<Location:> C:\Program Files\Microsoft Visual Studio\MyProjects\treact <default, OK>
Check <Create new workspace>, <Platforms:> Win32, <OK>

(3) Copy source files

Copy all source files (as mentioned above in Steps 2, 4 and 5 but in subdirectory
~/integer not/integer*8) to the defaults location, or,
C:\Program Files\Microsoft Visual Studio\MyProjects\treact

(4) Add source files

Select <Project>, <Add to project>, <Files...>
Select all source files (a total of 18 files):

(5) Compiler settings

Select <Project>, <Settings...>
<Settings for>: Win32 Release
<General>, <Not Using MFC>
<Intermediate:> blank
<Output files:> blank

<Fortran>
<Category>: General
<Optimization level:> Optimize for Speed
In the <Project option> window, add the following option:
/real_size:64 (in fact, IMPLICIT DOUBLE PRECISION (A-H, O-Z) statements

are coded in the source files, without this option the executable code
is still working, but this option does not result in additional
computing time)
In the <Project option> window, delete the following option:

/check:bounds
Leave all other settings unchanged

<Link>
<Output File Name:> treact.exe (or other convenient name for executable)
<OK>

(6) Compile and Link

<Build>, <Build treact.exe>
After this step, file treact.exe will be found in ~\MyProjects\treact\debug

(7) Miscellaneous

A new executable must be built for each EOS module.

To redimension TOUGH2, adjust parameter statements in file T2 and chempar23.inc:
<File>, <Open...>
<File name:> T2 or chempar23.inc
Modify parameter statements, save file, and rebuild treact.

---

3. EXECUTION

3.1. Execution

Step 1 Create a directory in user's computer.

Step 2: Prepare a total of four input files (or copy files from sample problems and then modify): flow.inp, solute.inp, chemical.inp, and chemical database input (user specified name).

Step 3 Type: treact, if this executable file is in the current directory.

3.2. Test

Users should run several of the sample problems to check on proper code installation. Due to machine-dependent roundoff, TOUGHREACT may produce slightly different results on different computers. For same-size time steps, all primary variables (pressure P, temperature T, saturation S, etc.) and their changes (DX1, DX2,...) should agree to typically four digits or better (file flow.out). However, on different computers the iteration sequence for a time step may be slightly different because of roundoff, and occasionally a different number of iterations may be required for convergence. If automatic time stepping is used, a different number of iterations for convergence may subsequently cause different-size time steps to be taken; naturally this will then produce somewhat larger discrepancies in results because of different time truncation errors.

Of all the numbers processed by TOUGHREACT, the most sensitive are the residuals, i.e. the differences between left-hand sides (accumulation terms) and right-hand sides of the governing equations. During the Newton-Raphson iteration process these residuals are reduced to smaller and smaller values, until they drop below specified convergence tolerances. As convergence is approached, the residuals are subject to increasingly severe numerical cancellation, arising from subtracting two numbers with diminishing difference. Maximum residuals are (optionally) printed in file "flow.out" during the iteration process as "MAX. RES.", and are also printed in the header of a full time step printout (as "MAX. RES." or "RERM"). These numbers can serve as a convenient check when evaluating reproducibility of

code applications. Small numerical differences due to roundoff etc.
will first show up in different values for “MAX. RES.”, long before
giving any visible changes in primary variables or their increments.

****

TOUGHREACT is documented in:

T. Xu, E. Sonnenthal, N. Spycher, and K. Pruess, TOUGHREACT Userís Guide: A Simulation Program for Non-isothermal Multiphase Reactive Geochemical Transport in Variably Saturated Geologic Media, Lawrence Berkeley National Laboratory Report LBNL-55460, July 2004.

TOUGH2 V2 is documented in:

K. Pruess, C. Oldenburg, and G. Moridis, TOUGH2 User’s Guide, Version 2.0, Lawrence Berkeley National Laboratory Report LBNL-43134, November 1999.

****

Distribution of the TOUGHREACT code is handled by

Energy Science and Technology Software Center
P.O. Box 1020
Oak Ridge, Tennessee 37831
U.S.A.

phone (865) 576-2606
fax (865) 576-6436
email: estsc@adonis.osti.gov
WorldWideWeb: http://www.osti.gov/estsc/

The code development team can be reached as follows

TIANFU XU
Mail Stop 90-1116
Lawrence Berkeley National Lab.
Berkeley, CA 94720
U.S.A.

fax (510) 486-5686
email: Tianfu_Xu@lbl.gov

Users are requested to tell us about any bugs that may be encountered.
We also like to hear about code improvements and enhancements; send
e-mail to Tianfu_Xu@lbl.gov.

****
* Additional information is available on the TOUGH2 homepage on the web *
*   *
*   http://www-esd.lbl.gov/TOUGH2/   *
****

.....
.....READ.ME.....READ.ME.....READ.ME.....READ.ME.....
.....
.....Update 10/04.....
.....